

10 Jahre FAI Projekt

Thomas Lange
Institut für Informatik, Universität zu Köln
Pohligstr. 1
50969 Köln
Deutschland
`lange@informatik.uni-koeln.de`

1 Zusammenfassung

Vor zehn Jahren wurde Version 1.0 der Software FAI (Fully Automatic Installation) veröffentlicht. Damals war FAI eine automatisierte Installationsprozedur für ein Cluster. Heute ist FAI ein universelles Werkzeug, das neben Installation und Softwareverteilung auch das Konfigurationsmanagement für verschiedene Linux Distributionen umfasst. Dabei wird die gesamte Computerinfrastruktur betrachtet und nicht einzelne Rechner, die es zu administrieren gilt. Im Laufe der letzten zehn Jahre hat sich das Tool stetig weiterentwickelt und ist heute als wichtiges Werkzeug für Administratoren etabliert.

In diesem Artikel wird die Funktionsweise von FAI erklärt. Anschließend vergleichen wir FAI kurz mit anderen Tools. Am Ende wird die 10-jährige Geschichte des Projekts beschrieben.

2 Überblick FAI

Im Namen Fully Automatic Installation wird schon ausgedrückt, was die wichtigen Eigenschaften ausmachen. FAI soll die Installation von Computern automatisiert ausführen. Dies soll vollautomatisch geschehen, sodass anschließend keine zusätzlichen manuellen Arbeitsschritte notwendig sind. Dadurch muss der Administrator nur noch den Installationsprozess anstoßen und der Rechner steht nach kurzer Zeit betriebsbereit für den Anwender zur Verfügung. FAI umfasst neben der Grundinstallation des Betriebssystems auch die Installation der Anwendungssoftware und die Konfiguration aller Komponenten nach den Vorgaben des Administrators. Außerdem unterstützt FAI das Konfigurationsmanagement nach der Erstinstallation. Eine Änderung der Konfiguration kann während des Betriebs von zentraler Stelle aus gesteuert auf den Rechnern ausgeführt werden.

2.1 Wie FAI funktioniert

FAI automatisiert den Installationsprozess, indem dieser nach Vorgaben automatisiert abläuft. Dabei werden die gleichen Schritte wie bei einer Installation von Hand ausgeführt. FAI benutzt kein Image (Master Image, Golden Image) zur Installation wie einige andere Tools.

Install Client wird der Rechner genannt, der neu aufgesetzt wird. Auf ihm selbst laufen die FAI Skripte. Die notwendigen Netzwerkdienste für den Bootprozess und die Installation werden vom FAI Installationsserver bereit gestellt. Er liefert allen Clients die Beschreibung, wie die Installation abläuft. Die Beschreibung der Installation ist in mehrere Verzeichnisse und Dateien aufgeteilt. Dieses Dateisystem wird Config Space genannt.

Die Clients beziehen ihre Softwarepakete aus einem Paket Depot, z.B. ein Debian Spiegel, der auch auf dem Installationsserver liegen kann. Dieser benötigt für die Pakete der i386 Architektur für Debian 5.0 (aka lenny) ca. 24 GB.

2.2 Schritte einer FAI Installation

Die normalen Schritte einer Installation sind:

- Starte ein Linux System, ohne die lokale Festplatte zu nutzen
- Partitioniere die Festplatte und erzeuge die Dateisysteme
- Installiere eine Auswahl von Softwarepaketen
- Passe die Software an die eigenen Bedürfnisse an
- Starte das neue System von der lokalen Festplatte

Bei einer Installation von Hand (z.B. mit dem Debian Installer) muss der Administrator einige Fragen beantworten. Sie beschreiben wie die Platte partitioniert werden soll, welche Software ausgewählt wird und wie diese Software konfiguriert werden soll. Je mehr Einstellungsmöglichkeiten der Administrator haben möchte, desto mehr Fragen hat er zu beantworten. Trotzdem kann immer nur eine Grundkonfiguration der Software mit diesen Fragen erzielt werden. Für eine genauere, feingliedrigere Anpassung werden nach der Grundinstallation meist viele Dateien von Hand angepasst oder ein Konfigurationsmanagement aufgerufen. FAI verbindet diese beiden Schritte und bietet in allen Teilen eine große Flexibilität.

2.3 Konfigurationsdateien

Installationsroutinen wie Jumpstart (Solaris) oder Kickstart (Red Hat) beschreiben in einer einzigen Datei für jeden Rechner, wie die automatische Installation ablaufen soll. Für größere und umfangreichere Installationsszenarien entwickeln viele Administratoren zusätzliche Mechanismen um diese Konfigurationsdateien dynamisch zu generieren. Dies kann im Falle von Kickstart auf der Seite eines Webserver gemacht werden, da die Kickstart Konfigurationsdatei vom Client via HTTP empfangen werden kann.

FAI teilt die Konfigurationsinformationen auf mehrere Dateien auf, die für je einen Teilschritt der Installation gelten. So wird die Gesamtkonfiguration aus den einzelnen Teilen automatisch zusammengesetzt, und der Administrator muss keinen eigenen Mechanismus zum Erzeugen der Konfiguration programmieren.

Für alle Schritte der Installation und Konfiguration nutzt FAI einfache Textdateien, die beschreiben wie die Installation ablaufen soll. Für die Festplattenpartitionierung wird eine an die `fstab` angelehnte Konfigurationsdatei benutzt. Diese kann jeder Administrator intuitiv ohne Spezialwissen mit einem Texteditor erzeugen. Abbildung 1 zeigt ein kurzes Beispiel.

```
# <type> <mountpoint> <size>    <fs type> <mount options> <misc options>

disk_config disk1

primary /                4G-6G    ext3    rw,noatime,errors=remount-ro
logical swap             1G-1500 swap    rw
logical /tmp              1G-15%   ext3    rw,noatime,nosuid
logical /home             100-     ext3    rw,noatime,nosuid,nodev
```

Abbildung 1: `disk_config/demohost`: Beispiel der Plattenpartitionierung

Hier wird die erste Festplatte (egal ob `sda`, `hda`, oder `hde`) partitioniert. Besonders elegant ist dabei die Angabe der gewünschten Partitionsgröße. So wird die erste Partition zwischen 4 und 6 Gbytes groß, abhängig von der Größe der gesamten Platte und davon wie groß die anderen Partitionen werden sollen. Die letzte Partition wird mindestens 100 Mbytes groß, wird aber den restlichen freien Platz bekommen. Zusätzlich werden in dem Beispiel Optionen für das Einhängen des Dateisystems angegeben. So wird z.B. die Partition `/home` mit den Optionen `rw`, `noatime`, `nosuid`, `nodev` in die `fstab` eingetragen. Unterstützt werden die Dateisysteme vom Typ `ext2`, `ext3`, `ext4`, `vfat`, `xfs`, `reiserfs` und `ntfs`, bei denen auch zusätzliche Feineinstellungen angegeben werden können. Um z.B. ein Label und den reservierten

Platz für den Administrator anzugeben (5% der Partitionsgröße) und automatische Dateisystemprüfungen abzuschalten wird folgendes hinzugefügt:

```
createopts="-L var -m 5" tuneopts="-c 0 -i 0"
```

Es ist auch möglich Software RAID und LVM Partitionen in dieser Konfiguration zu beschreiben. Das FAI interne Perl-Programm `setup-storage` liest diese Konfiguration ein und führt daraufhin die Partitionierung mit Hilfe von `parted` aus (siehe Abbildung 10). Dann konfiguriert es optional Software RAID und LVM und erstellt die Dateisysteme und die Datei `/etc/fstab`.

Für die Paketauswahl wird ein anderes Format genutzt. Die Zeile `PACKAGES` (in Abbildung 2) definiert mit welchem Befehl die darauf folgenden Softwarepakete installiert werden. Neben `aptitude`, `apt-get` werden auch andere Paketmanager wie z.B. `smart`, `yum`, `zypper` und andere unterstützt. Es ist ebenso möglich Pakete im Nachhinein zu deinstallieren oder eine ganz bestimmte Version zu installieren. Abhängigkeiten innerhalb der Pakete werden wie üblich von den Paketmanagern aufgelöst.

```
PACKAGES aptitude
fai-client cfengine2
debconf-utils file
less linuxlogo
openssh-client openssh-server

# spezielle Pakete fuer i386 Hardware
PACKAGES aptitude i386
linux-image-486
mentest86+
```

Abbildung 2: `package_config/FAIBASE`: Einfaches Beispiel der Paketauswahl

Diese beiden Beispiele sollen zeigen, wie leicht es ist eine Konfiguration für eine FAI Installation zu erstellen. Es wird bewusst kein XML eingesetzt (wie es AutoYaST für SUSE macht), da der Fokus darauf gerichtet ist, dass der Administrator die Konfiguration per Hand erstellen kann, ohne ein gesondertes Programm zu benötigen.

Softwarepakete werden mit der Standardkonfiguration installiert. Unter Debian kann diese durch den Preseeding Mechanismus [13] verändert werden. Dieser besitzt aber nur eine beschränkte Konfigurationsmöglichkeit für die Pakete. Bei RPM Paketen und unter Solaris gibt es dies gar nicht. In den meisten Fällen wird die exakte Anpassung der Software durch das Ausführen von weiteren Skripten realisiert. Diese Skripte sind je nach Anforderungen an die Konfiguration sehr aufwendig. FAI unterstützt an dieser Stelle das Ausführen von beliebig vielen Skripten, die in unterschiedlichen Skriptsprachen geschrieben sein können. So sind neben Shell, Perl und expect [14] auch `cfengine` Skripte möglich. Diese Skripte werden durch das FAI Klassensystem (siehe Kapitel 3.4) in eine logische Struktur gebracht. Sowohl Solaris, SUSE, Red Hat und der Debian Installer mit Preseeding bieten an dieser Stelle nur das Ausführen eines Skriptes an. Daher schreiben viele Administratoren einfache Erweiterungen, um diese Beschränkung zu umgehen.

3 Technische Details

Üblicherweise wird eine FAI Installation über das Netzwerk durchgeführt. Dabei erhält der Client alle Informationen und Daten von dem Installationsserver. Aus einer bestehenden Netzwerkinstallation kann durch zwei Befehle (`fai-mirror` und `fai-cd`) ein bootbares Installationsmedium erzeugt werden. Damit kann dann von CD-ROM oder USB Stick die gleiche Installation ausgeführt werden wie über das Netzwerk.

3.1 Aufsetzen des Installationsservers

Ein Installationsserver kann mit wenigen Kommandos auf einem Debian System eingerichtet werden. Da nur übliche Netzwerkdienste für eine FAI Installation benötigt werden, kann er auch auf jedem anderen Unix-artigen System laufen. Ein FAI Installationsserver besteht aus folgenden Komponenten:

Netzwerkdienste: DHCP, TFTP und NFS

NFS-Root: Das Dateisystem für die Clients

Config Space: Er besteht aus mehreren Konfigurationsdateien, die in Unterverzeichnissen verteilt sind

Optional kann noch ein Spiegel aller Debianpakete auf dem Server verwendet werden, um den Zugriff auf die Softwarepakete im lokalen Netz zu beschleunigen.

Die Install Clients booten via PXE von der Netzwerkkarte. Dafür sind TFTP und DHCP Dienste notwendig. Es können vorhandene Server im Netz genutzt werden, oder man installiert sie auf dem Installationsserver. Zum Exportieren der NFS-Root ist ein NFS Server obligatorisch. Für alle Dienste enthält FAI fertige Konfigurationen.

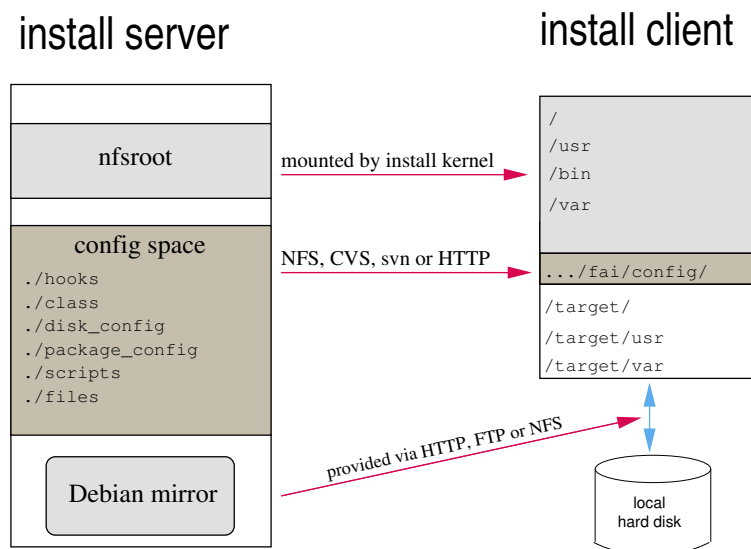


Abbildung 3: Zusammenspiel Server und Client

Am einfachsten kann der FAI Server aufgesetzt werden, indem das Debian Paket `fai-quickstart` installiert wird und anschließend der Befehl `fai-setup -v` ausgeführt wird. Dies erzeugt die NFS-Root und kopiert die Beispiele für eine FAI Konfiguration an die richtige Stelle. Die NFS-Root wird von den Clients via NFS eingebunden und fungiert als Root Dateisystem während der Installation. Der optionale Debian Spiegel kann mit dem Befehl `debmirror` sehr einfach angelegt werden.

3.2 Client Informationen auf dem Server

Vor dem Booten muss von jedem Install Client die MAC Adresse, IP Adresse und der Hostname in die `dhcpd.conf` und `/etc/hosts` Datei eingetragen werden. Folgende Zeilen sollten in der DHCP Konfiguration eingefügt werden:

```
use-host-decl-names on;  
server-name faiserver;  
next-server faiserver;  
filename "fai/pxe/linux.0";  
  
host demohost {hardware ethernet 0:2:43:A5:F5:44;fixed-address demohost;}
```

Zum Erzeugen der PXELINUX [12] Konfiguration, die beim Booten des Client benötigt wird, wird der Befehl `fai-chboot` genutzt. Um eine Konfiguration für den Rechner `demohost` zu erzeugen, die eine Installation ausführt und auf den Config Space via NFS vom Installationsserver mit Namen `faiserver` zugreift, nutzt man folgenden Befehl:

```
> fai-chboot -IFv -u nfs://faiserver/srv/fai/config demohost
demohost has 134.95.1.213 in hex 865F01D5
Writing file /srv/tftp/fai/pxelinux.cfg/865F01D5 for demohost
```

Daraus wird folgende PXE Konfigurationsdatei erzeugt:

```
# generated by fai-chboot for host demohost with IP 134.95.1.213
default fai-generated

label fai-generated
kernel vmlinuz-2.6.26-2-686
append initrd=initrd.img-2.6.26-2-686 ip=dhcp root=/dev/nfs nfsroot=/srv/fai/nfsroot \
boot=live FAI_FLAGS=verbose,sshd,createvt \
FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config FAI_ACTION=install
```

3.3 Booten des Clients

Wenn der Install Client von der Netzwerkkarte mit PXE bootet, werden zuerst vom DHCP Server die notwendigen Netzwerkparameter zugewiesen. Danach wird die Datei `pxelinux.0` und dann eine Konfiguration für PXELINUX via TFTP geladen. Darin wird der Kernel und das initial RAM filesystem (`initramfs`) mit zusätzlichen Kernelparametern angegeben. Dann werden das Kernel Image und das `initramfs` via TFTP übertragen und gebootet. Der Kernel erhält sein ganzes Root Dateisystem nun via NFS. Dies ist die auf dem Server erzeugte NFS-Root. Diese wird nur Read-Only gemountet, doch mit Hilfe des Overlay-Dateisystems aufs[3, 11] (Another Unionfs) wird es für den Client schreibbar. Somit läuft nun ein vollständiges Linux ohne Benutzung der lokalen Platte auf dem Install Client und die FAI Installation wird gestartet. Abbildung 4 zeigt den Ablauf leicht vereinfacht.

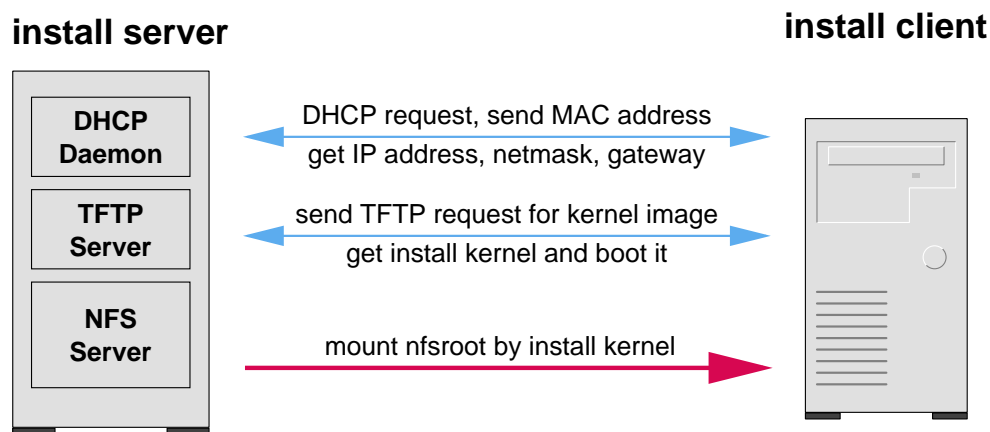


Abbildung 4: Bootprozess

Nun kann das FAI Skript seine Arbeit aufnehmen. Zuerst wird der Config Space, der über die Variable `FAI_CONFIG_SRC` definiert ist, auf dem Client verfügbar gemacht. Üblicherweise wird dazu ein Verzeichnis via NFS vom Installationsserver eingehängt. Es gibt aber auch die Möglichkeit, über andere Protokolle auf den Config Space zuzugreifen. Unterstützt werden zur Zeit verschiedene Varianten von `cvs`, `svn`, `http` und `git`. Abbildung 5 zeigt den Ausschnitt aus der wichtigen Logdatei `fai.log`, der den Anfang der FAI Installation zeigt:

```

-----
Fully Automatic Installation - FAI

FAI 3.3.5, 05 Apr 2010    (c) 1999-2009
Thomas Lange <lange@informatik.uni-koeln.de>
-----

Calling task_confdir
Kernel currently running: Linux 2.6.26-2-686 i686 GNU/Linux
Kernel parameters: initrd=initrd.img-2.6.26-2-686 ip=dhcp root=/dev/nfs
nfsroot=/srv/fai/nfsroot boot=live FAI_FLAGS=verbose,sshd,createvt FAI_ACTION=install
FAI_CONFIG_SRC=nfs://faiserver/srv/fai/config BOOT_IMAGE=vmlinuz-2.6.26-2-686
Reading /tmp/fai/boot.log
FAI_FLAGS: verbose sshd createvt
FAI_CONFIG_SRC is set to nfs://faiserver/srv/fai/cslenny
Configuration space faiserver:/srv/fai/config mounted to /var/lib/fai/config

```

Abbildung 5: Anfang der FAI Installation des Rechners demohost

3.4 Das FAI Klassensystem

Nachdem nun der Config Space auf dem Client verfügbar ist, führt FAI alle Skripte im Unterverzeichnis `class/` aus (siehe Abbildung 6), deren Namen mit zwei Ziffern beginnen. Alles was diese Skripte in die Standardausgabe schreiben wird als Klassenname von FAI definiert. So entsteht eine einfache, geordnete Liste von Klassennamen, die gleichzeitig eine Priorität von niedrig nach hoch definiert. Klassen, die zuerst ausgegeben werden, haben eine niedrigere Priorität als die zuletzt genannten. Klassennamen werden üblicherweise in Großbuchstaben geschrieben mit Ausnahme des Hostnamen, der auch Klassenname ist. Über die Zuordnung von Rechnern zu Klassen ist das Gruppieren von Rechnern möglich, die im weiteren Verlauf gewisse Teile der Konfigurationsinformationen miteinander teilen. Sie müssen nicht alle Konfigurationsinformationen teilen, sondern können zum Beispiel nur die Art der Plattenpartitionierung gemeinsam nutzen.

```

fai-class: Defining classes.
Executing /var/lib/fai/config/class/10-base-classes.
10-base-classes      OK.
Executing /var/lib/fai/config/class/50-host-classes.
50-host-classes      OK.
List of all classes:  DEFAULT I386 FAIBASE CSMAIN GERMAN GRUB demohost LAST

```

Abbildung 6: Klassendefinition im `fai.log`

```

#!/bin/bash

# assign classes hosts

# use a list of classes for our demo machine
case $HOSTNAME in
    demohost)
        echo "FAIBASE CSMAIN GERMAN" ;;
    gnomehost)
        echo "FAIBASE DHCP DEMO XORG GNOME" ;;
    atom*)
        echo "FAIBASE DHCP CLUSTER" ;;
    *)
        echo "FAIBASE DHCP" ;;
esac

(ifclass I386 || ifclass AMD64) && echo GRUB

```

Abbildung 7: Das Skript `class/50-host-classes`

Abbildung 7 zeigt wie man anhand des Rechnernamens (vom DHCP zugewiesen) verschiedene Klassen zuordnen kann. Allen Rechner deren Hostname mit `atom` beginnt, werden die Klassen `FAIBASE`,

DHCP und CLUSTER zugeordnet. Die Rechner demohost und gnomehost bekommen andere Klassen zugewiesen. Alle Rechner, die in der Klasse I386 oder AMD64 enthalten sind (diese wurde durch das Skript 10-base-classes vorher zugewiesen) erhalten zusätzlich die Klasse GRUB. Die Einfachheit dieser Technik für die Klassenzuordnung macht deren Flexibilität aus. Wer seine Klassenzuordnungen in einer Datenbank speichern möchte, schreibt nur ein kurzes Skript das die Liste der Klassennamen für einen Rechner auf der Standardausgabe ausgibt. Dies kann anhand von IP Adresse, Rechnernamen oder irgend einer anderen Information des Client geschehen. Es gibt auch Ansätze, kompliziertere Klassensysteme mit Subklassen oder Abhängigkeiten untereinander zu nutzen. FAI bietet dies nicht direkt an, eröffnet aber über die einfache Programmierung der Klassenzuordnung jedem die Option, dies zu nutzen.

Im weiteren Verlauf der Installation nutzen alle Teile das Klassensystem. So werden verschiedene Variablen über Dateien definiert, die als Dateinamen einen Klassennamen und den Suffix `.var` besitzen. Abbildung 8 zeigt in der Logdatei, wie Variablen über die Dateien `FAIBASE.var`, `CSMAIN.var` und `GERMAN.var` definiert werden. Dabei wird Shell Syntax genutzt.

```
Calling task_defvar
Executing FAIBASE.var
++ FAI_ALLOW_UNSIGNED=1
++ CONSOLEFONT=
++ KEYMAP=us-latin1
++ UTC=yes
++ TIMEZONE=Europe/Berlin
++ USE_SETUP_STORAGE=1
Executing CSMAN.var
++ ROOTPW=' $1$DE4.iseK$dlF229r2mm33x13YswMkI3'
++ YPDMAIN=MY.NIS
++ YPSVR=nisserv
Executing GERMAN.var
++ KEYMAP=de-latin1-nodeadkeys
```

Abbildung 8: Definition von Variablen im `fai.log`

3.5 Partitionierung der Festplatten

Auch die Plattenpartitionierung benutzt das Klassensystem. Dazu wählt FAI eine Konfigurationsdatei aus dem Unterverzeichnis `disk_config/` des Config Space aus. Ausgewählt wird die Datei, die den Klassennamen mit der höchsten Priorität besitzt, der für den Client definiert ist. In diesem Fall (siehe Abbildung 10) ist dies die Datei `demohost`. Zwar sind auch die Klassen `FAIBASE` und `CSMAIN` für diesen Client definiert und es gibt entsprechende Dateien in diesem Verzeichnis (siehe Abbildung 9), jedoch ist `demohost` die Klasse mit der höchsten Priorität, da sie am weitesten hinten in der Liste der Klassen steht (Abbildung 6).

```
disk_config/
|-- FAIBASE
|-- CSMAN
|-- WIKI
|-- demohost
|-- paeffgen
`-- rievkoche
```

Abbildung 9: Dateien im Unterverzeichnis `disk_config/`

Die Einfachheit dieses Klassensystems besteht also darin, dass eine neue Klasse dadurch erstellt wird, indem eine Konfigurationsdatei mit dem Klassennamen in dem entsprechenden Unterverzeichnis erzeugt wird. Dann muss die Klasse nur noch einem oder mehreren Rechnern zugeordnet werden. Eine bereits existierende Klasse z.B. `DESKTOP`, die nur eine Liste von Softwarepaketen definiert, kann

```

Starting setup-storage 1.2.1
Using config file: /var/lib/fai/config/disk_config/demohost
Executing: parted -s /dev/sda mklabel msdos
Executing: parted -s /dev/sda mkpart primary ext3 32256B 5362882559B
Executing: parted -s /dev/sda mkpart extended 5362882560B 10149995519B
Executing: parted -s /dev/sda mkpart logical linux-swaps 5362914816B 6933911039B
Executing: parted -s /dev/sda mkpart logical ext3 6933943296B 8003197439B
Executing: parted -s /dev/sda mkpart logical ext3 8003229696B 10149995519B
Executing: mkfs.ext3 -m15 /dev/sda1
Executing: mkswap /dev/sda5
Executing: mkfs.ext3 -m1 /dev/sda6
Executing: mkfs.ext3 -m0 /dev/sda7
Calling task_mountdisks
Enable swap device /dev/sda5
Mounting /dev/sda1 to /target/
Mounting /dev/sda7 to /target/files/scratch
Mounting /dev/sda6 to /target/tmp

```

Abbildung 10: Festplatten Partitionierung

zusätzlich um eine Plattenpartitionierung erweitert werden, indem eine Datei `DESKTOP` im Verzeichnis `disk_config/` erzeugt wird. Alle Rechner, die zur Klasse `DESKTOP` gehören nutzen dann automatisch auch diese Konfiguration.

3.6 Softwarepakete

Im nächsten Schritt der FAI Installation werden Softwarepakete anhand von Klassen ausgewählt und installiert. Dazu werden alle Dateien (im Gegensatz zu einer Datei bei der Plattenpartitionierung) im Unterverzeichnis `package_config/`, die einen Klassennamen haben in die Auswahl mit einbezogen. Dies zeigt der folgende Ausschnitt der `fai.log` Datei:

```

install_packages: reading config files from directory /var/lib/fai/config/package_config
install_packages: read config file DEFAULT
install_packages: read config file FAIBASE
install_packages: read config file CSMAIN
install_packages: read config file demohost
install_packages: executing chroot /target apt-get clean
install_packages: executing chroot /target aptitude -R -y install memtest86+ grub lilo-
cpufrequtils cfengine2 cron hdparm less linuxlogo nfs-common nscd rdate rsync
openssh-client openssh-server iceweasel iceweasel-l10n-de icedove icedove-l10n-de

```

Das neue Dateisystem wurde zuvor unter `/target` eingehängt und nachdem dort ein minimales System erzeugt wurde, können nun via `chroot` innerhalb dieser Umgebung die Softwarepakete mit `aptitude` installiert werden. Die Liste wird aus allen Einträgen der zuvor eingelesenen Konfigurationsdateien erzeugt. In diesen sind einfache logische UND Verknüpfungen über mehrere Klassen möglich, was hier nicht weiter erläutert wird.

3.7 Skripte zur Anpassung

Nachdem alle Softwarepakete installiert sind, können nun Skripte für das Konfigurationsmanagement aufgerufen werden. Damit für jede Klasse mehrere Skripte in unterschiedlichen Skriptsprachen ausgeführt werden können, werden im Config Space im Unterverzeichnis `scripts/<KLASSENNAME>` alle Skripte ausgeführt, die mit zwei Ziffern beginnen. Die Reihenfolge wird durch die Zahlen am Anfang bestimmt. Abbildung 11 zeigt den Ablauf etwas verkürzt im `fai.log`.

Diese Skripte machen üblicherweise die notwendigen Anpassungen an Dateien in `/etc`. Man kann dabei einfache Shell Skripte nutzen, oder ein vorhandenes Konfigurationsmanagement. Sinnvoll ist ein Mix aus verschiedenen Skriptsprachen, da so jede ihre Stärken ausspielen kann. Für häufige Aufgaben helfen die zwei FAI Befehle `ainsl` und `fcopy`, die im Folgenden erläutert werden.


```

Calling task_configure
Executing shell: FAIBASE/10-misc
Executing shell: FAIBASE/30-interface
Executing cfagent: FAIBASE/40-misc
Executing shell: CSMMAIN/11-nis
Executing cfagent: CSMMAIN/20-misc
Executing shell: CSMMAIN/21-net
Executing shell: CSMMAIN/31-sendmail
Executing shell: CSMMAIN/40-sound
Executing cfagent: CSMMAIN/91-nvidia
Executing shell: GRUB/10-setup
Executing shell: GRUB/30-win-boot
Executing shell: LAST/50-misc

```

Abbildung 11: Aufruf der Skripte zur lokalen Anpassung

Eine zu installierende `xorg.conf` Datei kann aus einer Menge von vorbereiteten Templates elegant mit dem Befehl `fcopy` installiert werden. Dazu wählt `fcopy` aus der Liste der Klassen das Template mit der höchsten Priorität aus und kopiert es an die richtige Stelle. Es reicht der kurze Befehl:

```
fcopy /etc/X11/xorg.conf
```

Dieser Befehl ist für alle Clients gleich und bedarf keiner weiteren Fallunterscheidung.

Der Befehl `ainsl` (angelehnt an `AppendIfNoSuchLine`, eine Funktion in `cfengine`) kann eine Zeile an eine Datei anhängen, wenn diese Zeile noch nicht in der Datei vorkommt. Somit kann ohne den Einsatz von `cfengine` diese praktische Funktion genutzt werden.

3.8 Ende der Installation

Am Ende der FAI Installation werden noch einige Informationen ausgegeben (siehe Abbildung 12), wie z.B. die Installationszeit. Der Install Client teilt nun dem Server mit, dass die Installation beendet ist und deaktiviert seine eigene PXE Konfiguration auf dem Server. Somit bekommt er beim folgenden Neustart eine neue PXE Konfiguration, die den Client dann von lokaler Festplatte starten lässt. Abschließend speichert der Client alle Logdateien der Installation auf dem Installationsserver. Mit Config Space und den Logdateien liegen nun alle relevanten Informationen der Installation auf dem Server. So hat man eine automatische Dokumentation der gesamten Installation.

```

Filesystem      1K-blocks      Used Available Use% Mounted on
rootfs          1033568        3456   1030112   1% /
134.95.9.149:/srv/fai/nfsroot
                 309571072 116958208 177033216  40% /live/image
aufs            1033568        3456   1030112   1% /
faiserver:/srv/fai/config
                 309571072 116958208 177033216  40% /var/lib/fai/config
/dev/sda1       5154852      3079220   1290060  71% /target
/dev/sda7       2063504       35876   2027628   2% /target/home
/dev/sda6       1027768       17688    999640   2% /target/tmp

Fri Apr 16 10:03:26 UTC 2010
The install took 721 seconds.
Calling task_chboot
disable pxe config for 134.95.1.213 in hex 865F01D5
Source hook: savelog.LAST.source
ERRORS found in log files. See /tmp/fai/error.log
savelog.LAST.source OK.
Calling task_savelog
Save log files via rsh to fai@faiserver:demohost//install-20100415_082124

```

Abbildung 12: Ende einer FAI Installation

3.9 Installationszeiten

Die Zeit für eine Installation wird hauptsächlich durch die Menge der zu installierenden Software bestimmt. Da viele Schreiboperationen auf der Festplatte geschehen, hat die Geschwindigkeit der Festplatte auch einen Einfluss. Abbildung 13 listet einige Beispiele auf. Die kurzen Installationszeiten zeigen auch, das FAI sehr gut als Disaster Recovery Tool eingesetzt werden kann. Innerhalb kürzester Zeit ist ein Rechner wieder komplett hergestellt, ohne das ein Systemadministrator alle technischen Details beherrschen muss.

CPU	MHz	RAM	Software	Zeit
Pentium 4	3000	1024MB	300 MB	105 sec
Pentium 4	3000	1024MB	1100 MB	6 min
Pentium 4	3000	1024MB	2200 MB	10 min
Athlon	800	512MB	300 MB	4 min
Athlon	800	512MB	1100 MB	17 min
Athlon	800	512MB	2200 MB	32 min
Pentium III	850	256MB	180 MB	3 min
PentiumPro	200	128MB	800 MB	28 min

Abbildung 13: Installationszeiten

Installiert man ein große Anzahl von Rechnern gleichzeitig, kann man mit einfachen Mitteln erreichen, dass die Hauptlast des Netzwerkverkehrs (die Verteilung der Softwarepakete) auf mehrere Rechner verteilt wird. Durch einfache Round Robin Mechanismen wird der Zugriff aller Clients auf mehrere Server verteilt. Auch wenn NFS für die NFS-Root eingesetzt wird, ist dies kein limitierender Faktor, da alle Clients nur lesend zugreifen und im Vergleich zu den Softwarepaketen wenige Daten übertragen werden. Somit kann die NFS-Root auf nur einem Server liegen. Der Config Space selber ist sehr klein und kann deshalb in der Betrachtung unberücksichtigt bleiben.

4 Sonstige Eigenschaften

Es gibt noch viele weitere Funktionen in FAI, die hier nicht erläutern werden. Einige Eigenschaften sollen aber noch kurz erwähnt werden.

Während der ganzen Installation hat der Administrator Remote-Zugriff auf jeden Client über eine SSH Verbindung. Somit kann er die Installation beobachten und Fehler aus der Ferne erkennen, beheben und die Installation neu starten.

FAI begann als Installationswerkzeug für Debian. Mittlerweile werden auch Ubuntu und RPM basierte Linux Distributionen unterstützt. Auch die Installation von virtuellen Rechnern mit verschiedenen Techniken wie Xen [2], Vserver, VirtualBox, qemu oder das Aufsetzen einer chroot Umgebung und das Erzeugen einer Live-CD beherrscht FAI. Das ISO Image der GRML Live-CD [10] wird mit Hilfe von FAI vollautomatisch erzeugt.

Neben der Erstinstallation kann FAI auch zum täglichen Konfigurationsmanagement eingesetzt werden. Dies wird `softupdate` genannt.

Wenn mehrere Clients gleichzeitig installiert werden, hilft der FAI Monitor Daemon mit seiner GUI und zeigt an, in welchem Teil der Installation sich ein Client befindet und ob die vorherigen Schritte erfolgreich abgeschlossen wurden. Abbildung 14 zeigt ein Beispiel.



Abbildung 14: GUI des FAI Monitor Daemons

5 Vergleich mit anderen Tools

FAI wird manchmal mit anderen Tools verglichen wie z.B. cfengine, bcfg2 oder Puppet. Dies sind jedoch Werkzeuge, die nur das Konfigurationsmanagement übernehmen. Sie können erst eingesetzt werden, wenn auf dem Rechner das Betriebssystem und einige grundlegende Softwarepakete installiert sind. FAI kann diese Werkzeuge zur Anpassung der Konfiguration nutzen. Dem Administrator bleibt dabei aber freigestellt welches er einsetzt, oder ob er mehrere davon kombiniert. Das FAI Klassensystem hilft ihm dabei, Ordnung in diese Skripte zu bekommen.

FAI macht mehr als reine Installationswerkzeuge, wie der Debian Installer mit Preseeding (D-I), Jumpstart, Kickstart oder AutoYaST. Diese dienen nur der Erstinstallation und sind von dem Konfigurationsmanagement völlig losgelöst. FAI ist auch das einzige Tool, das verschiedene Linux Distributionen installieren kann.

Image basierte Tools (z.B. clonezilla, dd, partimage, systemimager) sind oft nicht vollautomatisch und können nur einen vorher (per Hand) installierten Rechner klonen. Bei einigen Tools ist die Festplattenpartitionierung nur per Hand möglich, oder Software RAID wird nicht unterstützt. Außerdem kann mit einem Image nur ein Typ von Rechnern installiert werden. Unterschiede in der Hardware, Software oder Konfiguration erfordern meist ein neues Image. Dies skaliert nicht, wenn viele unterschiedliche Systeme installiert werden sollen.

6 Geschichte des FAI Projekts

6.1 Anfänge von FAI

Ende 1998 kaufte ein Lehrstuhl für Informatik der Universität zu Köln ein Beowulf Cluster. Es bestand aus einem Server und 16 Rechner mit je zwei Pentium II Prozessoren mit 400 MHz und je 256 MB RAM und 4 GB Festplatte¹. Als Netzwerk wurde ein 100 MBit/s Fast Ethernet mit einen 24 Port Switch realisiert. Diese Ausstattung entsprach der üblichen Leistungsfähigkeit zu dieser Zeit.

Der Autor hatte als Systemadministrator die Installation und den Betrieb des Clusters zu organisieren. Durch die langjährige Erfahrung mit Solaris Jumpstart[9] war die Idee geboren etwas Vergleichbares für eine Linux Installation zu nutzen. Das Cluster sollte mit Debian GNU/Linux installiert werden. Da für Debian noch keine automatische Installationsroutine existierte, begann ein dreiköpfiges Team mit der Entwicklung einer solchen Installation. Während der mehrmonatigen Entwicklung bis zur ersten

¹Einer dieser Rechner dient noch heute (2010) als Dateiserver in der Arbeitsgruppe

Version entstand der Name FAI. FAI steht für Fully Automatic Installation und verdeutlicht, dass nicht nur ein Teil des Installationsprozesses sondern alle Aufgaben des Administrators bei der Installation und Konfiguration automatisch ausgeführt werden.

Die Erfahrung mit Solaris Jumpstart zeigte an vielen Stellen einen starren und nicht erweiterbaren Ablauf der Installation. Deshalb wurde bei der Entwicklung von FAI besonderer Wert auf Flexibilität und einfache Erweiterbarkeit gelegt. FAI enthielt von Anfang an das Klassenkonzept. Diese Idee wurde von den Erweiterungen[8] von Casper Dik für Solaris Jumpstart inspiriert.

Die erste Präsentation von FAI erfolgte am 8. September 1999 auf dem Linux Kongress in Augsburg in Form eines halbtägigen Tutorials. Im Anschluss an einen Vortrag über FAI konnten die Teilnehmer ihr Wissen direkt in die Tat umsetzen und die Software an Rechnern vor Ort ausprobieren.

Am 21. Dezember 1999 wurde dann die Version 1.0 angekündigt. In einem Tar Archiv verpackt unterstützte sie Debian 2.1 (aka slink) und die Kernelversion 2.0.36. Zu dieser Zeit bestand FAI aus knapp 1700 Zeilen Shell und Perl Code. Davon fielen alleine über 900 Zeilen auf das Partitionierungstool. Die aktuelle Version 3.3.5 besteht aus über 12000 Zeilen Code, immer noch wenig im Vergleich zu anderer Software. Version 1.0 beinhaltete auch den Technical Report [1], der die Arbeitsweise von FAI dokumentierte.

Schon in der ersten Version der TODO Datei (siehe Abbildung 15), war der wichtigste Punkt, andere Linux Distributionen zu unterstützen.

```
TODO list for FAI, december 20, 1999

1. Most people wish that we would support other Linux distributions
   (redhat, Suse)
2. Booting a kernel with pcmcia support for automatic installation of
   notebooks and laptops
3. Cleanup the code of some perl scripts
4. install/class/S*.{pl,sh} scripts should check and resolve
   dependencies
5. Make some change in rcS to run with Debian 2.2 and futher releases
6. class/S*.source: define variabel tftplink which is the name of the
   link created in /tftplink. At present it's always clusterimage
```

Abbildung 15: TODO von FAI 1.0

Im Januar 2000 gab es mit dem Projekt FAI4RPM [16] den ersten Versuch RPM Pakete mit FAI zu installieren. Version 0.2 mit Namen pyFAI wurde dann teilweise in Python geschrieben, blieb jedoch die letzte Version dieses Projektes. Mittlerweile kann FAI aber RPM Pakete installieren und unterstützt auch die Paketmanager verschiedener Distributionen.

Anfang 2000 wurde Version 1.0.1 veröffentlicht, und im April 2000 wurde eine eigene Mailingliste für FAI eingerichtet. Diese Liste wurde das wichtigste Kommunikationsmittel, wenn es um Diskussionen und Ankündigungen rund um FAI geht. Nachdem der Autor den offiziellen Status des Debian Entwicklers (Debian Developer, DD) erlangt hatte, erschien in Juni 2000 FAI Version 1.1beta1 als erstes offizielles Debian Paket. Eine Woche später wurde dann ein Fork von FAI angekündigt mit Namen NAIS [15]. Dieser Fork wurde von zwei Studenten, die an FAI mitentwickelt hatten gestartet. Außer einer ersten Version 0.22beta1.0 wurde aber keine weitere Version in der Folgezeit herausgebracht. Dies zeigte, dass ein Softwareprojekt nicht nur technische Probleme lösen muss, sondern auch das Abwandern von Entwicklern oder die Konkurrenz durch ein anderes Projekt Einfluss auf das eigene Projekt haben kann. Zur Zeit gibt es noch das Projekt lucie [17], das einige Teile von FAI 3.2.17 enthält. Es implementiert zusätzlich einige FAI Funktionen in Ruby, indem es Shell Kommandos aus einem Ruby Skript heraus aufruft.

6.2 Dokumentation, Vorträge, Messen, Treffen

2005 startete das Wiki für FAI [5]. Es enthält viele zusätzliche Informationen von Benutzern zu FAI, die aber typischerweise nicht alle gepflegt werden. Hierzu gab es schon Diskussionen wie man dies

verbessern könnte, oder ob man veraltete Informationen besser entfernen sollte.

Ungefähr zur gleichen Zeit trafen sich auch die ersten FAI Entwickler und Anwender im IRC in einem eigenen Kanal. Der IRC Channel `#fai` auf `irc.debian.org` ist mittlerweile ein sehr wichtiges Kommunikationsmittel, in dem gerade neue Nutzer schnell Hilfe erhalten.

Neben vielen Vorträgen von Thomas Lange haben auch andere Entwickler und Nutzer weltweit über FAI berichtet [7]. Mehr als 50 Vorträge, Paper und Artikel in Zeitschriften wie iX, Linux Magazin und IT-Administrator sind im Laufe der Jahre dabei zusammengekommen. Eine kleine Auswahl der Konferenzen, auf denen Vorträge über FAI gehalten wurden sind:

Linux-Kongress, LinuxTag, LCA, SANE, FOSDEM, SUCON, CeBIT, FrOSCon, DebConf, UKUUG.

Es gab mehrere Projekte in der Ausbildung zum Fachinformatiker, die sich mit FAI beschäftigt haben. Eine Diplomarbeit an der Hochschule Winterthur und weitere Arbeiten an Hochschulen befassten sich ebenso mit FAI. Neben englisch- und deutschsprachigen Dokumentationen gibt es auch noch einige Beiträge in Spanisch, Französisch, Tschechisch, Rumänisch und Portugiesisch. Leider wurden diese Dokumentationen nicht weiter gepflegt.

Bisher gibt es kein eigenes Buch über FAI, jedoch enthält das Buch über Xen Virtualisierung [2] ein umfangreiches Kapitel über die Nutzung von FAI zum Aufsetzen virtueller Rechner.

Außerdem war FAI mehrfach auf Konferenzen und Messen mit einem Projektstand vertreten. Dort wurde dem Besucher der Ablauf einer Installation vorgeführt und sie konnten sich bei den Entwicklern über FAI informieren.

In den vielen Jahren gab es einige Entwicklertreffen. Dabei wurde meist im Linuxhotel ein Wochenende lang intensiv an FAI gearbeitet. Diese Treffen sind immer sehr ergiebig und erleichtern den Austausch zwischen den Entwicklern.

6.3 Die Webseite

Leider hatte FAI sehr viele Jahre nur eine sehr einfache, textbasierte Webseite. Dies war für das Jahr 1999 sicherlich noch angemessen, jedoch wurde erst im Dezember 2009 die neue Webseite mit aktuellem Design und neuem Aufbau angekündigt. Abbildung 16 zeigt das alte und neue Webdesign.

Leider fehlt dem FAI Projekt immer noch ein Logo. Jedoch ist diese Lücke erkannt und wird hoffentlich bald gefüllt.

6.4 Der Fragebogen

Um detaillierte Rückmeldungen von FAI Anwendern zu erhalten, wurde der FAI Fragebogen entworfen. Die erste Antwort im Juni 2001 berichtete noch davon, dass wegen Kinderkrankheiten kein Rechner installiert wurde. Mittlerweile gibt es aber über 220 Antworten, in denen die Nutzer von kleinen und großen Installationen mit bis zu 3000 Rechnern berichten. Dies ist ein sehr guter Überblick [6] in welchen verschiedenen Bereichen FAI eingesetzt wird.

Einige Beispiele von FAI Anwendern sind: Opera Software (300 Cluster Knoten), Stadtverwaltung München (ca. 2000 Arbeitsplätze, geplant 14.000), BUF Compagnie (ca. 1000 Arbeitsplätze, Rendering und Storage Knoten), Albert Einstein Institut (1725 HPC Knoten), Mathworks (1000 Arbeitsplätze), Ziviti (260 Rechner auf IBM zSeries Z10), Spotify (Musik Anbieter, 450 Server für verschiedene Dienste, davon 150 virtuelle Server), ein Webportal (3000 Server), mobile.de (600 verschiedene Web-, Mail-, Datenbank- und Anwendungsserver). Ebenso verwenden viele Universitäten und Forschungseinrichtungen FAI zur Administration ihrer Arbeitsplätze, Server und Cluster.

Viele Dutzend Cluster für verschiedene Anwendungen wurden mit FAI aufgesetzt. Einige dieser Cluster schafften es sogar in die Top500.org, die Liste der schnellsten Computer der Welt. Eine Firma installierte weit in Deutschland verteilt Miniserver mit CF Speicher. Auch einige Schulungsräume werden mit FAI verwaltet, sodass sie vor Beginn einer Schulung automatisch neu installiert werden.

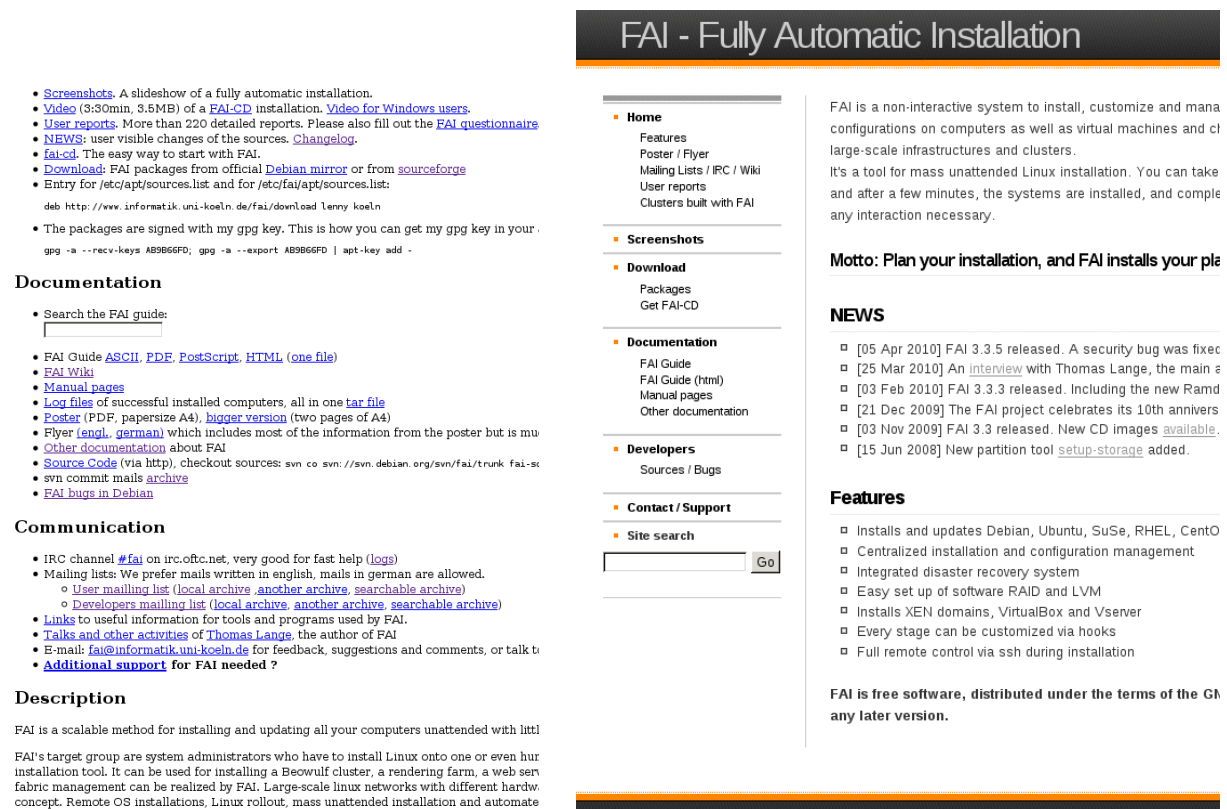


Abbildung 16: Alte und neue Webseite

6.5 Technische Entwicklung im Laufe der Zeit

In den letzten zehn Jahren wurde FAI immer weiter verbessert und erweitert. Dies ist hauptsächlich darauf zurückzuführen, dass Benutzer Rückmeldungen geben. Dadurch werden Fehler schneller erkannt, neue Funktionen werden gewünscht oder es werden sogar Erweiterungen mit einem Patch geschickt. FAI ist somit ein typisches Open Source Projekt, das sehr stark von den Benutzern mitgestaltet wird und dadurch so gut wurde.

Das Kommando zum Erzeugen der bootbaren FAI CD (mit nfsroot und Paketdepot) wurde erstmals mit Version 2.7 im Jahre 2005 ausgeliefert und kann seit 2007 auch bootbare USB Sticks erzeugen. Mit Version 3.0 wurde im September 2006 die Funktionalität hinzugefügt chroot Umgebungen zu erzeugen. Seit 2008 nutzt das Projekt GRML [10] (ein Debian-basiertes Live-System für Systemadministratoren) diese Funktion zum Bauen der verschiedenen CD/DVD Varianten. Somit ist es nun möglich täglich aktuelle ISO Dateien automatisch zu erzeugen.

Seit Version 3.2 arbeitet FAI mit dem Standard Debian Kernel mit initramfs. Dies erleichtert die Anpassung an neue Hardware, da kein speziell gebauter Kernel mehr benötigt wird, wie dies vorher der Fall war.

Nachdem schon einige Jahre lang die Idee bestand das Festplatten Partitionierungstool neu zu schreiben, um Software-RAID, LVM und GPT Unterstützung zu ermöglichen, wurde das neue Programm `setup-storage` endlich im Juni 2008 in FAI eingebaut. Dies ermöglicht nun auch Festplatten mit mehr als 2 TB zu partitionieren. Im Augenblick werden einige neue Features bzgl. LVM eingebaut.

In den zehn Jahren wurde FAI auf fast allen Hardwarearchitekturen eingesetzt. Neben den weit verbreiteten Architekturen wurden auch alte Alpha Rechner, ein Itanium Cluster, PowerPC und SPARC Rechner unter Solaris mit FAI installiert und gewartet. Selbst Mainframes sind mit zwei zSeries z10

Rechnern vertreten, auf denen FAI ca. 260 Debian Instanzen installiert hat.

Neben sehr unterschiedlicher Hardware wurde FAI aber auch in sehr unterschiedlichen Projekten genutzt. Dies ist eine der Stärken von FAI im Gegensatz zu vielen anderen Installations- und Konfigurationstools, die meist auf ein bestimmtes Nutzungsszenario und eine bestimmte Linux Distribution festgelegt sind. FAI hingegen arbeitet distributionsübergreifend. Es kann neben Debian auch Ubuntu, Red Hat, CentOS, SUSE und Mandriva installieren und wurde auch schon für Solaris auf SPARC genutzt.

7 Fazit

Das Projekt FAI ist in über zehn Jahren zu einem professionellen Installations- und Konfigurationswerkzeug herangewachsen. Seine Stärken liegen in der Flexibilität, Erweiterbarkeit und in der Fähigkeit unterschiedliche Linux Distributionen zu unterstützen. FAI kann dem Administrator viel Arbeit abnehmen, leider jedoch nicht die Planung der Installation. Ganz nach dem Slogan:

Plane deine Installation und FAI installiert deinen Plan

Literatur

- [1] Mattias Gärtner, Thomas Lange, Jens Rühmkorf: *The fully automatic installation of a Linux cluster*, Technical report 99.379, Institut für Informatik, Universität zu Köln, 1999
- [2] Henning Sprang, Timo Benk, Jaroslaw Zdrzalek, Ralph Dehner: *Xen: Virtualisierung unter Linux*, Open Source Press, München, 2006
- [3] Mirko Dölle: *Schicht um Schicht, Union-Dateisysteme unter Linux* in c't Magazin für Computertechnik, Heft 10, Seite 192ff, Heise Zeitschriften Verlag GmbH, Hannover, 2010
- [4] <http://www.informatik.uni-koeln.de/fai>
- [5] <http://faiwiki.informatik.uni-koeln.de>
- [6] <http://www.informatik.uni-koeln.de/fai/reports>
- [7] <http://www.informatik.uni-koeln.de/fai/otherdocs>
- [8] <http://www.science.uva.nl/pub/solaris/auto-install/>
- [9] <http://docs.sun.com/app/docs/doc/820-0179>
- [10] <http://www.grml.org>
- [11] <http://aufs.sourceforge.net>
- [12] <http://syslinux.zytor.com/pxe.php>
- [13] <http://d-i.alioth.debian.org/manual/de.i386/apb.html>
- [14] <http://expect.nist.gov>
- [15] <http://nais.sourceforge.net>
- [16] <http://www.mynetcologne.de/~nc-renardth/fai/>
- [17] <http://lucie.is.titech.ac.jp/trac/lucie/>